UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/637,408	08/07/2003	Sachin G. Deshpande	SLA1306	3156

50735 7590 07/17/2007

MADSON & AUSTIN
15 WEST SOUTH TEMPLE
SUITE 900
SALT LAKE CITY, UT 84101

EXAMINER	
VERDI, KIMBLEANN C	
ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
07/17/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/637,408	DESHPANDE, SACHIN G.	
	Examiner	Art Unit	
	Kacy Verdi	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *April 23, 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-33* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-33* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *December 29, 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This office action is in response to the Amendment filed on April 23, 2007. Claims 1-33 are pending in the current application. Applicants' arguments have been carefully considered, but they are not persuasive. Accordingly, this action has been made FINAL. All previously outstanding objections and rejections to the Applicant's disclosure and claims not contained in this Action have been respectfully withdrawn by the Examiner hereto.

Response to Amendment

1. Amendment to the specification overcomes the previous objections to the specification.

Amendment to claim 18 overcomes the 35 U.S.C. §101 rejection. Therefore, the rejection of claims 18-27 under 35 U.S.C. §101 is withdrawn.

Response to Arguments

2. Applicant's arguments filed on April 23, 2007 have been fully considered but they are not persuasive. In response to the Non-Final Office Action dated January 30, 2007, applicant argues in regards to claims 1, 18, and 28:

(1) **Lee does not teach or suggest "inserting an API level translator layer, wherein the API level translator layer does not require a stack for the second version of the protocol."**

(2) **Yu does not teach or suggest "inserting an API level translator layer, wherein the API level translator layer does not require a stack for the second version of the protocol."**

(3) **Yu does not teach or suggest "translating the function call to a translated function call wherein the translated function call uses raw sockets."** Yu recites that "the socket subroutines contained in host sockets library 97 serve as the application program interface (API) for TCP/IP. This API provides three types of communications services which use different components of TCP/IP. These are reliable stream delivery, connectionless datagram delivery and raw socket delivery." Yu, para. 7, lines 28-33

In response to argument (1), examiner respectfully disagrees and notes that Lee discloses inserting an API level translator layer, wherein the API level translator layer does not require a stack for the second version of the protocol. Lee teaches an API translator is inserted between TCP/IP module and network card driver (page 2, section 1., lines 7-8) using "Bump-in-the-Stack" (BIS) approach (page 2, section 1, lines 3-4), BIS is for systems with no IPv6 stack (page 3, section 2, lines 7-8). For example, no Ipv6 stack can be interpreted as the API level translator layer does not require a stack.

In response to argument (2), examiner respectfully disagrees and notes that Yu discloses inserting an API level translator layer, wherein the API level translator layer does not require a stack for the second version of the protocol. Yu teaches a socket interface layer which operatively connects through a TCP/IP network protocol stack (col. 2, lines 23-25). The socket server component in turn issues the appropriate host socket

library calls to the host socket library interface layer thereby eliminating both the need to communicate through additional protocol stacks and the need to provide additional communication hardware facilities (col. 2, lines 35-40). For example, the socket interface layer can be interpreted as the API level translator, since socket server components operatively connect ES TCP/IP application programs to the socket library interface layer of the host operating system in response to standard ES socket library calls issued by such programs; the socket command handler unit contains specific routines which map the ES socket library calls into appropriate input/output requests directed to the EMCU which in turn directs the requests to a main socket server component; and the socket server component in turn issues the appropriate host socket library calls to the host socket library interface layer (col. 2, line 27-37).

Overall, the socket mechanism of the present invention allows proprietary application programs running in the emulation environment access to host TCP/IP protocol stack communication facilities of the host enhanced UNIX operating system thereby eliminating the need to communicate through additional protocol stacks (col. 3, lines 11-16). For example, the socket layer eliminates the need to communicate through additional protocol stacks can be interpreted as the API level translator layer does not require a stack for the second version of the protocol.

In response to argument (3), examiner respectfully disagrees and notes that Yu discloses translating the function call to a translated function call wherein the translated function call uses raw sockets. Yu teaches the socket handler routing maps the ES Socket monitor call to an IORB (block 206, Fig. 2, col. 8, lines 27-29), the host socket

library provides API for raw socket delivery (col. 7, lines 29-33). For example mapping the ES Socket monitor call to an IORB can be interpreted as translating the function call to a translated function, the ES library call is applied as input to the ES Socket library (col. 8, lines 20-21). For example, the socket library provides API for raw socket delivery, can be interpreted as wherein the translated function call uses raw sockets.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-7, 10-14, 17-21, 24, 27-29, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Dual Stack Hosts using "Bump-in-the-API" (BIA) by Lee et al. (hereinafter Lee) in view of U.S. Patent No. 5,721,876 to Yu et al. (hereinafter Yu).

5. As to claim 1, Lee teaches the invention substantially as claimed including a method for a computer system for providing communication between a first system and a second system, wherein the first system uses a first version of a protocol and the second system uses a second version of the protocol, the method comprising:

providing a first application on the first system (IPv4 application Fig. 1, page 4);

inserting an API level translator layer (an API translator is inserted between TCP/IP module and network card driver, page 2, section 1., lines 7-8, using "Bump-in-the-Stack" (BIS) approach, page 2, section 1, lines 3-4), wherein the API level translator

layer does not require a stack for the second version of the protocol (BIS is for systems with no IPv6 stack, page 3, section 2, lines 7-8);

making a function call to a socket Application Programming Interface (API) for the first version (call IPv4 Socket API function Fig. 2, page 8);

translating the function call to a translated function call (Translate IPv4 into IPv6, page 2);

making a function call to the socket API for the translated function call (invoke IPv6 socket API function page 6, section 4.1, lines 23-24); and

passing a packet to a stack for the first version of the protocol (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15).

Lee does not explicitly teach wherein the translated function call uses raw sockets.

However, Yu teaches wherein the translated function call uses raw sockets (the host socket library provides API for raw socket delivery, col. 7, lines 29-33).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the host protocol stacks of Lee with the teachings of a Protocol stack and host Socket Library from Yu because these features would have provided a mechanism for proprietary application programs, utilizing communication network protocols, such as TCP/IP, implemented as part of the proprietary operating system (col. 1, lines 36-39 of Yu), to access host TCP/IP protocol stack communication

facilities, which in turn eliminates the need to communicate through additional protocol stacks (col. 3, lines 11-17 of Yu).

6. As to claim 2, Lee teaches the method of claim 1, wherein the first version of the protocol is IPv4 (Internet Protocol version 4) and wherein the second version of the protocol is IPv6 (Internet Protocol version 6) (section 2.1, page 3, lines 2-3).

7. As to claim 3, Lee teaches the method of claim 2, further comprising supplying IP headers (Address Mapper supplies destination addresses to header, Fig. 1).

8. As to claim 4, Lee teaches the method of claim 2, further comprising supplying IP headers only once (reply with one IPv4 address Fig. 2, page 7).

9. As to claim 5, Lee teaches the method of claim 2, wherein the method is implemented by inserting an API level translator layer between a socket API layer and a TCP/IPv4 layer (API translator, Fig. 1, section 1, page 2, lines 13-14).

10. As to claim 6, Lee as modified teaches the method of claim 2, wherein the method is implemented without using an IPv6 stack (TCP/IP Network Protocol Stack Facility 99, Fig. 1b of Yu).

11. As to claim 7, Lee teaches the method of claim 4, further comprising passing the packet to a network card driver (layer beneath TCP/IP module, section 1, page 2, line 7).

12. As to claim 10, Lee teaches the method of claim 2, further comprising using a Name Resolver service to perform name to address resolution related functions (Name Resolver, Fig. 1).

13. As to claim 11, Lee teaches the method of claim 10, wherein the Name Resolver service is configured to run on a separate host that includes an IPv4 stack and an IPv6 stack (Name Resolver, Fig. 1).

14. As to claim 12, Lee teaches the method of claim 11, wherein name to address resolution functions of the Name Resolver service use the IPv6 stack (Name Resolver, section 3.2, pages 4-5).

15. As to claim 13, Lee teaches the method of claim 12, wherein the Name Resolver service is further configured to receive a query from the first system, use the address resolution functions to obtain a record and send the record to the first system (Behavior of Originator, Fig. 2, page 7).

16. As to claim 14, Lee as modified teaches the method of claim 2, further comprising providing an alternate implementation for a sending-related IPv4 socket function, wherein the alternate implementation comprises:

intercepting an IPv4 socket API call to send a packet (Appendix A, API List Intercepted by BIA, page 13, line 32 of Lee);

translating the IPv4 socket API call (Translate IPv4 into IPv6, page 2 of Lee) to use a raw socket (raw socket delivery col. 7, lines 27-34 of Yu);

providing transport and IPv6 headers (Address Mapper supplies destination addresses to header, Fig. 1 of Lee);

calling a corresponding IPv4 socket API function for the raw socket call (invoke IPv6 socket API function page 6, section 4.1, lines 23-24 of Lee) (invoke socket function 200, Fig. 2, col. 8, lines 15-17 of Yu); and

passing the packet to the stack stack (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15 of Lee).

17. As to claim 17, Lee teaches the method of claim 14, further comprising passing the packet to a network card driver (TCP/IP module passes packet to next network card driver layer, which is the layer beneath TCP/IP module, section 1, page 2, line 7).

18. As to claim 18, Lee teaches the invention substantially as claimed including a set of executable instructions (for example socket API function calls) on a computer readable medium for providing communication between the IPv4 (Internet Protocol version 4) system and an IPv6 (Internet Protocol version 6) system, the instructions being configured to:

provide an IPv4 application on the IPv4 system (IPv4 application Fig. 1, page 3);

insert an API level translator layer (an API translator is inserted between TCP/IP module and network card driver, page 2, section 1., lines 7-8, using "Bump-in-the-Stack" (BIS) approach, page 2, section 1, lines 3-4), wherein the API level translator layer does not require a stack for the second version of the protocol (BIS is for systems with no IPv6 stack, page 3, section 2, lines 7-8);

make a function call to an IPv4 socket Application Programming Interface (API) (call IPv4 Socket API function Fig. 2, page 8);

translate the function call to a translated function call (Translate IPv4 into IPv6, page 2);

make another function call to the IPv4 socket API for the translated function call (invoke IPv6 socket API function page 6, section 4.1, lines 23-24); and

pass a packet to an IPv4 stack (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15).

Lee does not explicitly teach wherein the translated function call uses raw sockets.

However, Yu teaches wherein the translated function call uses raw sockets (the host socket library provides API for raw socket delivery, col. 7, lines 29-33).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the host protocol stacks of Lee with the teachings of a Protocol stack and host Socket Library from Yu because these features would have provided a mechanism for proprietary application programs, utilizing communication network protocols, such as TCP/IP, implemented as part of the proprietary operating system (col. 1, lines 36-39 of Yu), to access host TCP/IP protocol stack communication facilities, which in turn eliminates the need to communicate through additional protocol stacks (col. 3, lines 11-17 of Yu).

19. As to claim 19, Lee teaches the set of executable instructions (for example socket API function calls) of claim 18, wherein the method further comprises supplying IP headers only once (reply with one IPv4 address Fig. 2, page 7).

20. As to claim 20, Lee teaches the set of executable instructions (for example socket API function calls) of claim 18, wherein the method is implemented by inserting an API level translator layer between a socket API layer and a TCP/IPv4 layer (API translator, Fig. 1, section 1, page 2, lines 13-14).

21. As to claim 21, Lee as modified teaches the set of executable instructions (for example socket API function calls) of claim 18, wherein the method is implemented without using an IPv6 stack (TCP/IP Network Protocol Stack Facility 99, Fig. 1b of Yu).

22. As to claim 24, Lee as modified teaches the set of executable instructions (for example socket API function calls) of claim 18, wherein the method further comprises providing an alternate implementation for a sending-related IPv4 socket function, wherein the alternate implementation comprises:

intercepting an IPv4 socket API call to send data (Appendix A, API List Intercepted by BIA, page 13, line 32 of Lee);

translating the IPv4 socket API call (Translate IPv4 into IPv6, page 2 of Lee) to use a raw socket (raw socket delivery col. 7, lines 27-34 of Yu); providing transport and IPv6 headers (Address Mapper supplies destination addresses to header, Fig. 1 of Lee);

calling a corresponding IPv4 socket API function for the raw socket (invoke IPv6 socket API function page 6, section 4.1, lines 23-24 of Lee) (invoke socket function 200, Fig. 2, col. 8, lines 15-17 of Yu); and

passing the data to the IPv4 stack (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15 of Lee).

23. As to claim 27, Lee teaches the set of executable instructions (for example socket API function calls) of claim 24, wherein the method further comprises passing the packet to a network card driver (TCP/IP module passes packet to next network card driver layer, which is the layer beneath TCP/IP module, section 1, page 2, line 7).

24. As to claim 28, Lee teaches the invention substantially as claimed including a system for enabling an IPv4 (Internet Protocol version 4) application to communicate across a computer network using an IPv6 (Internet Protocol version 6) system, the system comprising:

wherein the executable instructions (for example socket API function calls) are configured to implement a method comprising:

inserting an API level translator layer (an API translator is inserted between TCP/IP module and network card driver, page 2, section 1., lines 7-8, using "Bump-in-the-Stack" (BIS) approach, page 2, section 1, lines 3-4), wherein the API level translator layer does not require a stack for the second version of the protocol (BIS is for systems with no IPv6 stack, page 3, section 2, lines 7-8);

making a function call to an IPv4 socket Application Programming Interface (API) (call IPv4 Socket API function Fig. 2, page 8);

translating the function call to a translated function call (Translate IPv4 into IPv6, page 2);

making another function call to the IPv4 socket API for the translated function call (invoke IPv6 socket API function page 6, section 4.1, lines 23-24; and

passing a packet to an IPv4 stack (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15).

Lee does not explicitly teach a computing device;

executable instructions executable on the computing device; and

wherein the translated function call uses raw sockets.

However, Yu teaches a computing device (Hardware Level 56, Operating System Level 64, and User Level 62, Fig. 1a);

executable instructions executable on the computing device (Hardware Level 56, Operating System Level 64, and User Level 62, Fig. 1a); and

wherein the translated function call uses raw sockets (the host socket library provides API for raw socket delivery, col. 7, lines 29-33).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have modified the host protocol stacks of Lee with the teachings of a Protocol stack and host Socket Library from Yu because these features would have provided a mechanism for proprietary application programs, utilizing communication network protocols, such as TCP/IP, implemented as part of the proprietary operating system (col. 1, lines 36-39 of Yu), to access host TCP/IP protocol stack communication facilities, which in turn eliminates the need to communicate through additional protocol stacks (col. 3, lines 11-17 of Yu).

25. As to claim 29, Lee teaches the system of claim 28, further comprising an API level translator layer between a socket API layer and a TCP/IPv4 layer (API translator, Fig. 1, section 1, page 2, lines 13-14).

26. As to claim 31, Lee as modified teaches the system of claim 28, wherein the method further comprises providing an alternate implementation for a sending-related IPv4 socket function, wherein the alternate implementation comprises:

intercepting an IPv4 socket API call to send the packet (Appendix A, API List Intercepted by BIA, page 13, line 32 of Lee);

translating the IPv4 socket API call (Translate IPv4 into IPv6, page 2 of Lee) to use a raw socket (raw socket delivery col. 7, lines 27-34 of Yu);

providing transport and IPv6 headers (Address Mapper supplies destination addresses to header, Fig. 1 of Lee);

calling a corresponding IPv4 socket API function for the raw socket (invoke IPv6 socket API function page 6, section 4.1, lines 23-24 of Lee) (invoke socket function 200, Fig. 2, col. 8, lines 15-17 of Yu); and

passing the packet to the IPv4 stack (IPv4 Application sends packet to stack when invoking socket API function page 6, section 4.1, lines 14-15 of Lee).

27. Claims 8, 22-23, and 30, are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee in view of Yu as applied to claims 2, 14, and 28 above, and further in view of U.S. Patent Application 2003/0165160 A1 to Minami et al. (hereinafter Minami).

28. As to claim 8, Lee as modified by Yu teaches the invention substantially as claimed including the method of claim 2, further comprising providing an alternate implementation for a reception-related IPv4 socket function, wherein the alternate implementation comprises:

receiving an incoming packet (e.g. data) on a raw socket (receive data from IPv6 host, Fig. 3, page 9 of Lee) (raw socket delivery col. 7, lines 27-34 of Yu);

Lee as modified by Yu does not explicitly teach checking a source host to determine the proper destination for the incoming packet;

checking a port number for the incoming packet;

stripping a transport and IP headers from the incoming packet; and

passing a payload (data) to a destination application.

However, Minami teaches checking a source host to determine the proper destination for the incoming packet (IP header field parsing module 2062, Fig. 20, paragraph [0280]-[0289]);

checking a port number for the incoming packet (parse destination port field of header, paragraph [0069]);

stripping a transport and IP headers from the incoming packet (Network Protocol Layer 101, Fig. 1, paragraph [0065]); and

passing a payload (data) to a destination application (passed to Data Handler 102, Fig. 1, paragraph [0065]).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the host system of Lee as modified by Yu with the teaching of a Gigabit Ethernet Adapter from Minami because this feature would have further provided a mechanism for decoding multiple network protocols in a byte-streaming manner concurrently and processes packet data in one pass, thereby reducing system memory and form factor requirements, while eliminating software CPU overhead (paragraph [0017]).

29. As to claim 22, Lee as further modified teaches the set of executable instructions (for example socket API function calls of Lee) of claim 18, wherein the method further comprises providing an alternate implementation for a reception-related IPv4 socket function, wherein the alternate implementation comprises:

receiving an incoming packet on a raw socket (receive data from IPv6 host, Fig. 3, page 9 of Lee) (raw socket delivery col. 7, lines 27-34 of Yu);

checking a source host to determine the proper destination for the incoming packet (IP header field parsing module 2062, Fig. 20, paragraph [0280]-[0289] of Minami);

checking a port number for the incoming packet (parse destination port field of header, paragraph [0069] of Minami);

stripping a transport and IP headers from the incoming packet (Network Protocol Layer 101, Fig. 1, paragraph [0065] of Minami); and

passing a payload (data) to the IPv4 application (using IPv4 function call to IPv4 Application Figure 3, page 9 of Lee) (raw socket delivery col. 7, lines 27-34 of Yu) (passed to Data Handler 102, Fig. 1, paragraph [0065] of Minami).

30. As to claim 23, Lee as modified teaches the set of executable instructions (for example socket API function calls of Lee) of claim 22, further comprising a computer-readable medium (memory 58b, Fig. 1a of Yu) for storing the executable instructions.

31. As to claim 30, Lee as further modified teaches the system of claim 28, wherein the method further comprises providing an alternate implementation for a reception-related IPv4 socket function, wherein the alternate implementation comprises:

receiving an incoming packet on a raw socket (receive data from IPv6 host, Fig. 3, page 9 of Lee) (raw socket delivery col. 7, lines 27-34 of Yu);

checking a source host to determine the proper destination for the incoming packet (IP header field parsing module 2062, Fig. 20, paragraph [0280]-[0289] of Minami);

checking a port number for the incoming packet (parse destination port field of header, paragraph [0069] of Minami);

stripping a transport and IP headers from the incoming packet (Network Protocol Layer 101, Fig. 1, paragraph [0065] of Minami); and

passing a payload (data) to the IPv4 application (using IPv4 function call to IPv4 Application Figure 3, page 9 of Lee) (raw socket delivery col. 7, lines 27-34 of Yu) (passed to Data Handler 102, Fig. 1, paragraph [0065] of Minami).

32. Claims 9, 15-16, 25-26, and 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lee in view of Yu as applied to claims 2, 14, 24, and 28 above, and further in view of "Transition Mechanisms for IPv6 Hosts and Routers" by R. Gilligan (hereinafter Gilligan).

33. As to claim 9, Lee as modified by Yu does not teach the method of claim 2, further comprising performing tunneling of IPv6 packets over IPv4 routing infrastructure.

However, Gilligan teaches performing tunneling of IPv6 packets over IPv4 routing infrastructure (Configured and Automatic tunneling of Ipv6 over IPv4, sections 4 and 5).

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to have further modified the host system of Lee as modified by Yu with the teachings of tunneling from Gilligan because this feature would have further

provided a mechanism for maintaining compatibility with IPv4 while deploying Ipv6 (section 1 Introduction, page 2, lines 3-4).

34. As to claim 15, Lee as further modified teaches the method of claim 14, further comprising performing tunneling of IPv6 packets over IPv4 routing infrastructure (Configured and Automatic tunneling of Ipv6 over IPv4, sections 4 and 5 of Gilligan).

35. As to claim 16, Lee as further modified teaches the method of claim 14, further comprising fragmenting the packet (Tunnel MTU and Fragmentation, section 3.2, page 11 of Gilligan).

36. As to claim 25, Lee as further modified teaches the set of executable instructions (for example socket API function calls of Lee) of claim 24, wherein the method further comprises performing tunneling of IPv6 packets over IPv4 routing infrastructure (Configured and Automatic tunneling of Ipv6 over IPv4, sections 4 and 5 of Gilligan).

37. As to claim 26, Lee as further modified teaches the set of executable instructions (for example socket API function calls of Lee) of claim 24, wherein the method further comprises fragmenting the packet (Tunnel MTU and Fragmentation, section 3.2, page 11 of Gilligan).

38. As to claim 32, Lee as further modified teaches the system of claim 28, wherein the method further comprises performing tunneling of IPv6 packets over IPv4 routing infrastructure (Configured and Automatic tunneling of Ipv6 over IPv4, sections 4 and 5 of Gilligan).

39. As to claim 33, Lee as further modified teaches the system of claim 28, wherein the method further comprises fragmenting the packet (Tunnel MTU and Fragmentation, section 3.2, page 11 of Gilligan).

Conclusion

40. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kacy Verdi whose telephone number is (571) 270-1654. The examiner can normally be reached on Monday-Friday 7:30am-5:00pm EST..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

June 27, 2007
KV

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER